

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Applicant(s): Vugranam C. Sreedhar
Docket No.: YOR920010262US2
Serial No.: 09/925,580
Filing Date: August 9, 2001
Group: 2191
Examiner: Ted T. Vo

Title: Method and Apparatus for Programming Software Components

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Applicant hereby replies to the non-final Office Action dated November 21, 2006, of claims 2-4, 10-12, and 18 of the above-identified patent application. A Request to Reinstate the appeal is submitted herewith. Appellant's original Appeal Brief in an Appeal of the final rejection of claims 2-4, 10-12, and 18 in the above-identified patent application was submitted on December 21, 2005.

REAL PARTY IN INTEREST

The present application is assigned to International Business Machines Corporation, as evidenced by an assignment recorded on August 9, 2001 in the United States Patent and Trademark Office at Reel 012067, Frame 0529. The assignee, International Business Machines Corporation, is the real party in interest.

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

STATUS OF CLAIMS

5 The present application was filed on August 9, 2001 with claims 1 through
20. Claims 1, 5-7, 9, 13-15, 17, and 19 were cancelled in the Amendment and Response
to Office Action dated August 26, 2004. Claims 8, 16, and 20 were cancelled in the
Amendment and Response to Office Action dated September 15, 2005. Claims 2-4, 10-
12, and 18 are presently pending in the above-identified patent application. Claims 2-4,
10 10-12, and 18 are rejected under 35 U.S.C. §102(b) as being anticipated by Posnak,
Lavender et al., "Adaptive Configuration an Object Structural Pattern for Adaptive
Applications," 1996 (hereinafter, "Lavender"). Claims 4, 12, and 18 are being appealed.

STATUS OF AMENDMENTS

15 The amendments in the Amendment and Response to Office Action dated
September 15, 2005 have been entered.

SUMMARY OF CLAIMED SUBJECT MATTER

20 Independent claim 4 is directed to a method executed by a processor (FIG.
2: 210) for programming a software component (FIG. 1: 110) comprising the steps of:
defining properties of the software component (FIG. 1: 110), including at least one input
port (FIG. 1: 112) and at least one output port (FIG. 1: 114; pages 4-5 and 10-11);
providing a software mechanism for instantiating the software component (pages 4-5);
and utilizing an attach command to attach at least one input port (FIG. 1: 112) to a class.
25 Each input port (FIG. 1: 112) in a component (FIG. 1: 110) should be attached (using the
attach command) to some concrete class with in the component (FIG. 1: 110). (Page 3,
lines 6-7, and page 12, lines 5-6.)

Independent claim 12 is directed to a system (FIG. 2: 200) for programming a software component (FIG. 1: 110) said system (FIG. 2: 200) comprising: a memory (FIG. 2: 220) that stores computer-readable code; and a processor (FIG. 2: 210) operatively coupled to said memory (FIG. 2: 220), said processor (FIG. 2: 210) configured to implement said computer-readable code, said computer-readable code configured to: define properties of the software component (FIG. 1: 110), including at least one input port (FIG. 1: 112) and at least one output port (FIG. 1: 114; pages 4-5 and 10-11); provide a software mechanism for instantiating the software component (pages 4-5); and utilize an attach command to attach at least one input port (FIG. 1: 112) to a class (page 3, lines 6-7, and page 12, lines 5-6).

Independent claim 18 is directed to an article of manufacture for programming a software component (FIG. 1: 110), said system (FIG. 2: 200) comprising: a computer readable medium having computer readable code means embodied thereon, said computer readable program code means comprising: a step to define properties of the software component (FIG. 1: 110), including at least one input port (FIG. 1: 112) and at least one output port (FIG. 1: 114; pages 4-5 and 10-11); a step to provide a software mechanism for instantiating the software component (pages 4-5); and a step to utilize an attach command to attach at least one input port (FIG. 1: 112) to a class (page 3, lines 6-7, and page 12, lines 5-6.)

STATEMENT OF GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 2-4, 10-12, and 18 are rejected under 35 U.S.C. §102(b) as being anticipated by Lavender.

ARGUMENT

Independent Claims 4, 12 and 18

Independent claims 4, 12, and 18 were rejected under 35 U.S.C. §102(b) as being anticipated by Lavender. Regarding claim 4, the Examiner asserts that Lavender

discloses utilizing an attach command to attach at least one of said at least one input port to a class.

Appellant notes that the “attach” in Lavendar *attaches an output port of one module to an input port of another module*. Lavendar does **not** utilize an attach command to *attach an input port to a class*. The present invention *utilizes an explicit attach command to attach an input port to a class*. For example, the present disclosure teaches that “a component *must attach each of its input ports* to a concrete class within it.” (Page 3, lines 3-4; emphasis added.) The present disclosure also teaches that “each input port 112 in a component 110 should be attached (*using the attach command*) to some concrete class with in the component 110.” (Page 12, lines 5-6; emphasis added.) Finally, the present disclosure teaches that

the following code segment defines a component 110, referred to as BooleanComp, implementing the template, BooleanTempl, defined above:

```
component BooleanComp implements BooleanTempl {
  attach xin BoolClass , // attach input port xin to class
  BoolClass ;
```

```
  BooleanComp() {...} // constructors.
  class BoolClass implements Bool {
    boolean not(boolean x) { . } ;
    boolean nand (boolean x, boolean y) {...}
  }
}
```

Thus, the input port xin is “attached” to the class

```
BoolClass
```

(Page 6, line 23, to page 7, line 6; emphasis added.)

Independent claims 4, 12, and 18 require utilizing an attach command to attach at least one of said at least one input port to a class.

Thus, Lavender does not disclose or suggest utilizing an attach command to attach at least one of said at least one input port to a class, as required by independent claims 4, 12, and 18.

Conclusion

The rejections of the cited claims under section 102 in view of Lavender et al. are therefore believed to be improper and should be withdrawn. The remaining rejected dependent claims are believed allowable for at least the reasons identified above
5 with respect to the independent claims.

The attention of the Examiner and the Appeal Board to this matter is appreciated.

Respectfully,

Kevin M. Mason

Date: February 21, 2007

Kevin M. Mason
Attorney for Applicant(s)
Reg. No. 36,597
Ryan, Mason & Lewis, LLP
1300 Post Road, Suite 205
Fairfield, CT 06824
(203) 255-6560

CLAIM APPENDIX

1 (Cancelled)

5 2. The method of claim 4, further comprising the step of allowing said software component to access an external environment only through said output port.

10 3. The method of claim 4, further comprising the step of allowing a client to access said software component only through said input port.

4. A method executed by a processor for programming a software component, said method comprising the steps of:

15 defining properties of said software component, including at least one input port and at least one output port;

providing a software mechanism for instantiating said software component; and

utilizing an attach command to attach at least one of said at least one input port to a class.

20

5. (Cancelled)

6. (Cancelled)

25 7. (Cancelled)

8. (Cancelled)

9. (Cancelled)

10. The system of claim 12, wherein said processor is further configured to allow said software component to access an external environment only through said output port.

5 11. The system of claim 12, wherein said processor is further configured to allow a client to access said software component only through said input port.

10 12. A system for programming a software component, said system comprising:
a memory that stores computer-readable code; and
a processor operatively coupled to said memory, said processor configured to implement said computer-readable code, said computer-readable code configured to:
define properties of said software component, including at least one input
15 port and at least one output port;
provide a software mechanism for instantiating said software component;
and
utilize an attach command to attach at least one of said at least one input
port to a class.

20

13. (Cancelled)

14. (Cancelled)

25

15. (Cancelled)

16. (Cancelled)

17. (Cancelled)

18. An article of manufacture for programming a software component, said system comprising:

5 a computer readable medium having computer readable code means embodied thereon, said computer readable program code means comprising:

a step to define properties of said software component, including at least one input port and at least one output port;

10 a step to provide a software mechanism for instantiating said software component; and

a step to utilize an attach command to attach at least one of said at least one input port to a class.

19. (Cancelled)

15

20. (Cancelled)

EVIDENCE APPENDIX

There is no evidence submitted pursuant to § 1.130, 1.131, or 1.132 or entered by the Examiner and relied upon by appellant.

RELATED PROCEEDINGS APPENDIX

There are no known decisions rendered by a court or the Board in any proceeding identified pursuant to paragraph (c)(1)(ii) of 37 CFR 41.37.